CS 540 Introduction to Artificial Intelligence
Games I

Fred Sala
University of Wisconsin-Madison

April 1, 2021

Announcements

- **Homeworks:**
  - None!
- **Midterm**: grading nearly done.
- Class roadmap:

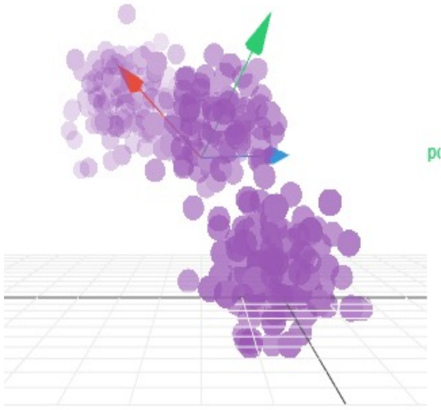
Thursday, April 1	**Games I**
Tuesday, April 6	Games II
Thursday, April 8	Search I
Tuesday, April 13	Search II

Artificial Intelligence

Outline

- Introduction to game theory
  - Properties of games, mathematical formulation
- Simultaneous Games
  - Normal form, strategies, dominance, Nash equilibrium
- Sequential Games
  - Game trees, minimax, search approaches

So Far in The Course



Victor Powell

We looked at techniques:

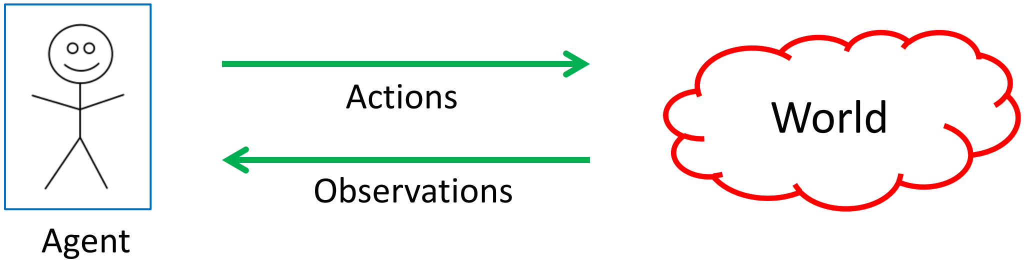
- **Unsupervised:** See data, do something with it. Unstructured.
- **Supervised:** Train a model to make predictions. More structure.
  - Training: as taking actions to get a reward
- **Games**: Much more structure.

indoor

outdoor

More General Model

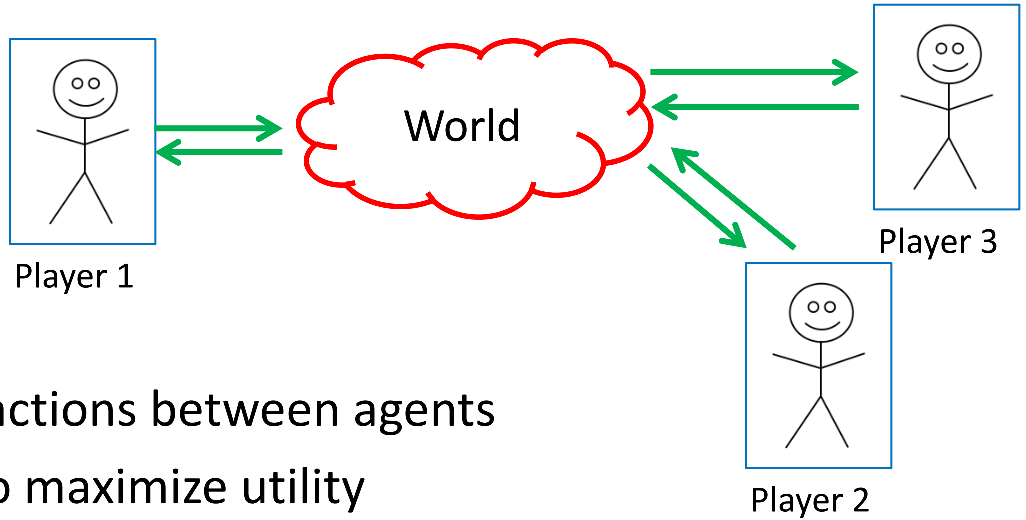
Suppose we have an **agent interacting** with the **world**



- Agent receives a reward based on state of the world
  - **Goal**: maximize reward / utility **($$$)**
  - Note: now **data** consists of actions & observations
  - Setup for decision theory, reinforcement learning, planning

Games: Multiple Agents

Games setup: **multiple** agents



- Now: interactions between agents
- Still want to maximize utility
- **Strategic** decision making.

Modeling Games: Properties

Let's work through **properties** of games

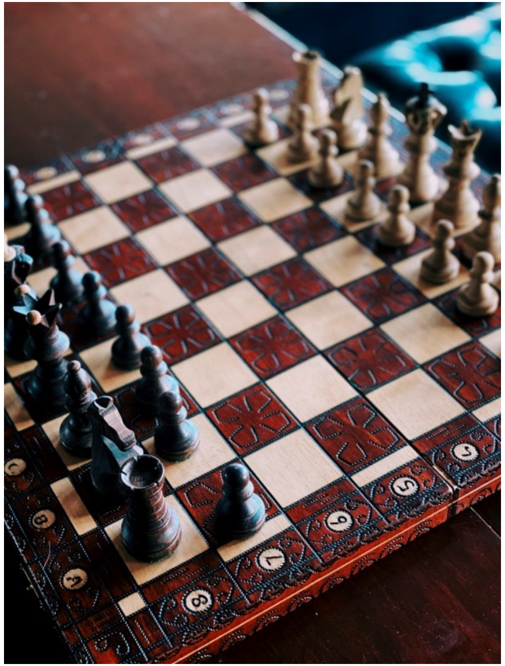
- **Number** of agents/players
- State & action spaces: **discrete** or **continuous**
- **Finite** or **infinite**
- **Deterministic** or **random**
- **Sum**: zero or positive or negative
- **Sequential** or **simultaneous**

Wiki

Property 1: **Number** of players

Pretty clear idea: 1 or more players

- Usually interested in ≥ 2 players
- Typically a finite number of players



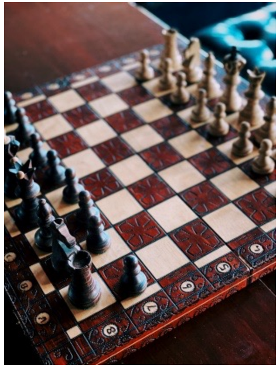
Property 2: Discrete or Continuous

Let's work through **properties** of games

- Recall the **world**. It is in a particular state, from a set of states
- Similarly, the actions the player takes are from an action space
- How big are these spaces? Finite, countable, uncountable?

Property 3: Finite or Infinite

Let's work through **properties** of games



- Most real-world games **finite**
- Lots of single-turn games; end immediately
  - Ex: rock/paper/scissors
- Other games' rules (state & action spaces) enforce termination
  - Ex: chess under FIDE rules ends in at most 8848 moves
- **Infinite example:** pick integers. First player to play a 5 loses

Property 4: Deterministic or Random

Let's work through **properties** of games

- Is there **chance** in the game?
- Note: randomness enters in different ways

Property 5: **Sums**

Let's work through **properties** of games

- **Sum**: zero or positive or negative
- Zero sum: for one player to win, the other has to lose
  - No "value" created

Red \ Blue	*A*	*B*	*C*
1	30, −30	−10, 10	20, −20
2	−10, 10	20, −20	−20, 20

- Can have other types of games: positive sum, negative sum.
  - Example: prisoner's dilemma

Property 6: Sequential or Simultaneous

Let's work through **properties** of games

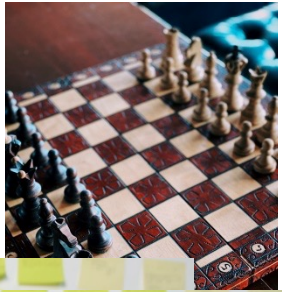
- Sequential or simultaneous
- Simultaneous: all players take action at the same time
- Sequential: take turns

- Simultaneous: players do not have information of others' moves. Ex: **RPS**
- Sequential: may or may not have **perfect information** (knowledge of all moves so far)

Examples

Let's apply this to examples:

1. Chess: 2-player, discrete, finite, deterministic, zero-sum, sequential (perfect information)
2. RPS: 2-player, discrete, finite, deterministic, zero-sum, simultaneous
3. Mario Kart: 4-player, continuous, infinite (?), random, zero-sum, simultaneous



Another Example: Prisoner's Dilemma

Famous example from the '50s.

Two prisoners A & B. Can choose to betray the other or not.

- A and B both betray, each of them serves two years in prison
- One betrays, the other doesn't: betrayer free, other three years
- Both do not betray: one year each

Properties: **2-player, discrete, finite, deterministic, negative-sum, simultaneous**

Why Do These Properties Matter?

Categorize games in different groups

- Can focus on understanding/analyzing/“solving” particular groups
- **Abstract** away details and see common patterns
- Understand how to produce a “good” overall outcome

How Does it Connect To Learning?

Obviously, learn how to play effectively

Also: suppose the players don't know something

- **Ex**: the reward / utility function is not known
- Common for real-world situations
  - How do we choose actions?
- Model the reward function and **learn it**
  - Try out actions and observe the rewards

Simultaneous Games

Simpler setting, easier to analyze

- Can express reward with a simple diagram
- Ex: for prisoner's dilemma

Player 2 Player 1	*Stay silent*	*Betray*
Stay silent	−1, −1	−3, 0
Betray	0, −3	−2, −2

Normal Form

Mathematical description of simult. games. Has:

- n players $\{1,2,\ldots,n\}$
- Player i strategy a_i from A_i. **All**: $a = (a_1, a_2, \ldots, a_n)$
- Player i gets rewards $u_i(a)$ for any outcome
  - **Note**: reward depends on other players!

- Setting: all of these spaces, rewards are **known**

Example of Normal Form

Ex: Prisoner's Dilemma

Player 2 Player 1	*Stay silent*	*Betray*
Stay silent	−1, −1	−3, 0
Betray	0, −3	−2, −2

- 2 players, 2 actions: yields 2x2 matrix
- Strategies: {Stay silent, betray} (i.e, binary)
- Rewards: {0,-1,-2,-3}

Dominant Strategies

Let's analyze such games. Some strategies are better

- Dominant strategy: if a_i better than a_i' *regardless* of what other players do, a_i is **dominant**
- I.e.,

$$u_i(a_i, a_{-i}) \geq u_i(a_i', a_{-i}) \forall a_i' \neq a_i \text{ and } \forall a_{-i}$$

All of the other entries of *a* excluding *i*

- Doesn't always exist!

Dominant Strategies Example

Back to Prisoner's Dilemma

- Examine all the entries: betray dominates
- Check:

Player 2 Player 1	*Stay silent*	*Betray*
Stay silent	−1, −1	−3, 0
Betray	0, −3	−2, −2

- Note: normal form helps **locate** dominant/dominated strategies.

Equilibrium

$a*$ is an equilibrium if all the players do not have an incentive to **unilaterally deviate**

$$u_i(a_i^*, a_{-i}^*) \geq u_i(a_i, a_{-i}^*) \quad \forall a_i \in A_i$$

- All players dominant strategies -> equilibrium
- Converse doesn’t hold (don’t need dominant strategies to get an equilibrium)

Pure and Mixed Strategies

So far, all our strategies are deterministic: "**pure**"

- Take a particular action, no randomness

Can also randomize actions: "**mixed**"

- Assign probabilities x_i to each action

$$x_i(a_i), \text{ where } \sum_{a_i \in A_i} x_i(a_i) = 1, x_i(a_i) \geq 0$$

- Note: have to now consider **expected rewards**

Nash Equilibrium

Consider the mixed strategy $x^* = (x_1^*, \ldots, x_n^*)$

- This is a **Nash equilibrium** if

$$u_i(x_i^*, x_{-1}^*) \geq u_i(x_i, x_{-i}^*) \quad \forall x_i \in \Delta_{A_i}, \forall i \in \{1, \ldots, n\}$$

Better than doing anything else, "**best response**"

Space of probability distributions

- Intuition: nobody can **increase expected reward** by changing only their own strategy. A type of solution!

Properties of Nash Equilibrium

Major result: (Nash ’51)

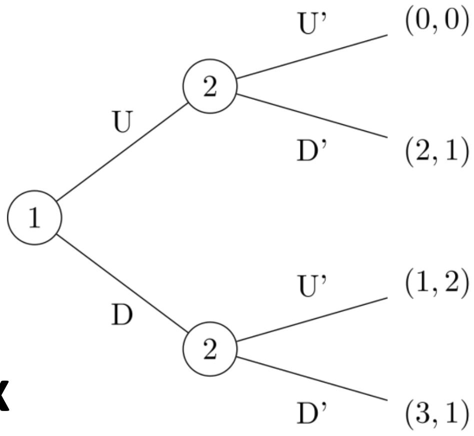
- Every finite game has at least one Nash equilibrium
  - But not necessarily **pure** (i.e., deterministic strategy)
- Could be more than one!
- Searching for Nash equilibria: computationally **hard**!

Example: rock/paper/scissors has (1/3, 1/3, 1/3) as a mixed strategy NE.

Sequential Games

More complex games with multiple moves

- Instead of normal form, **extensive form**
- Represent with a **tree**
- Perform search over the tree



Wiki

- Can still look for Nash equilibrium
  - Or, other criteria like **maximin** / **minimax**

II-Nim: Example Sequential Game

2 piles of sticks, each with 2 sticks.

- Each player takes one or more sticks from pile
- Take last stick: lose

(ii, ii)

- Two players: Max and Min
- If Max wins, the score is **+1**; otherwise **-1**
- Min's score is –Max's
- Use Max's as the score of the game

Game Trajectory

(ii, ii)

Game Trajectory

(ii, ii)

Max takes one stick from one pile

(i, ii)

Game Trajectory

(ii, ii)

Max takes one stick from one pile

(i, ii)

Min takes two sticks from the other pile

(i,-)

Game Trajectory

(ii, ii)

Max takes one stick from one pile

(i, ii)

Min takes two sticks from the other pile

(i,-)

Max takes the last stick

(-,-)

Max gets score **-1**

Game tree for II-Nim

Two players:
Max and Min

(ii ii) Max

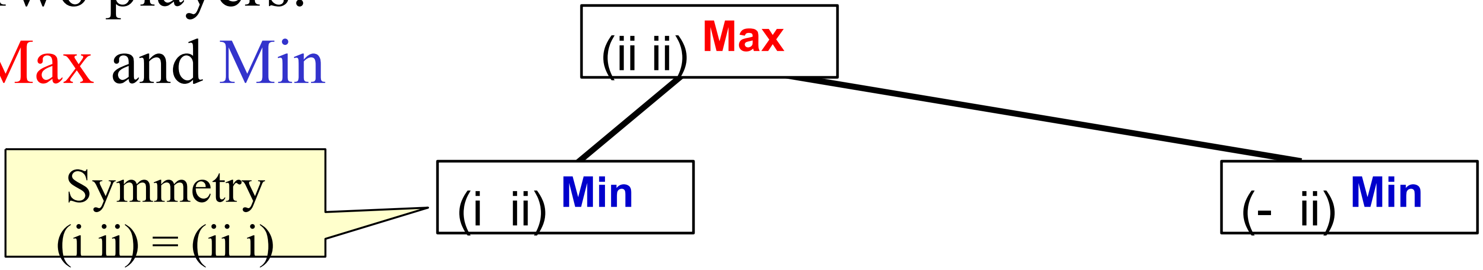
who is to move at this state

Convention: score is w.r.t. the first player Max. Min's score = – Max

Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

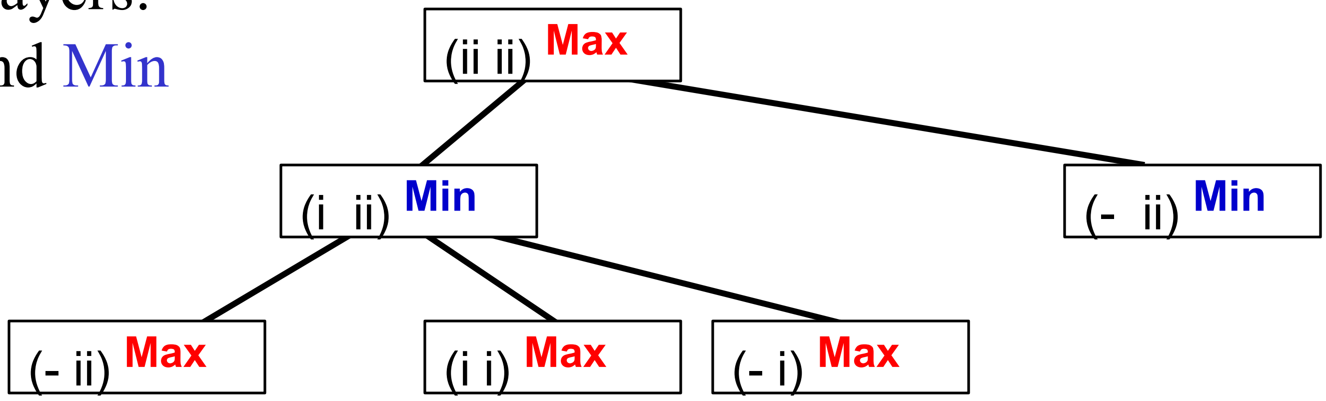
Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

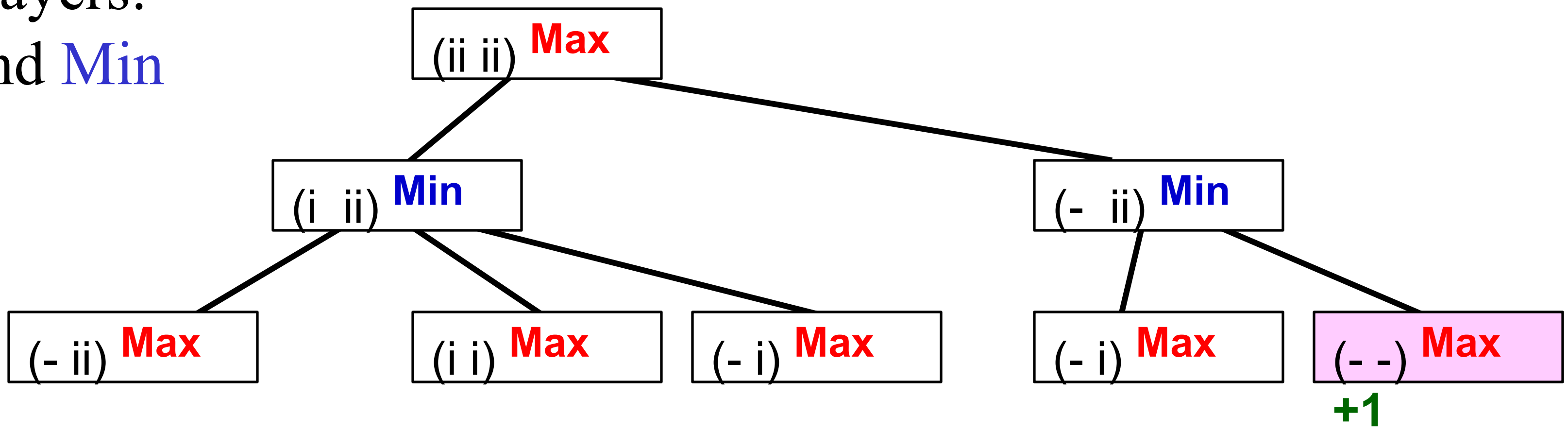
Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

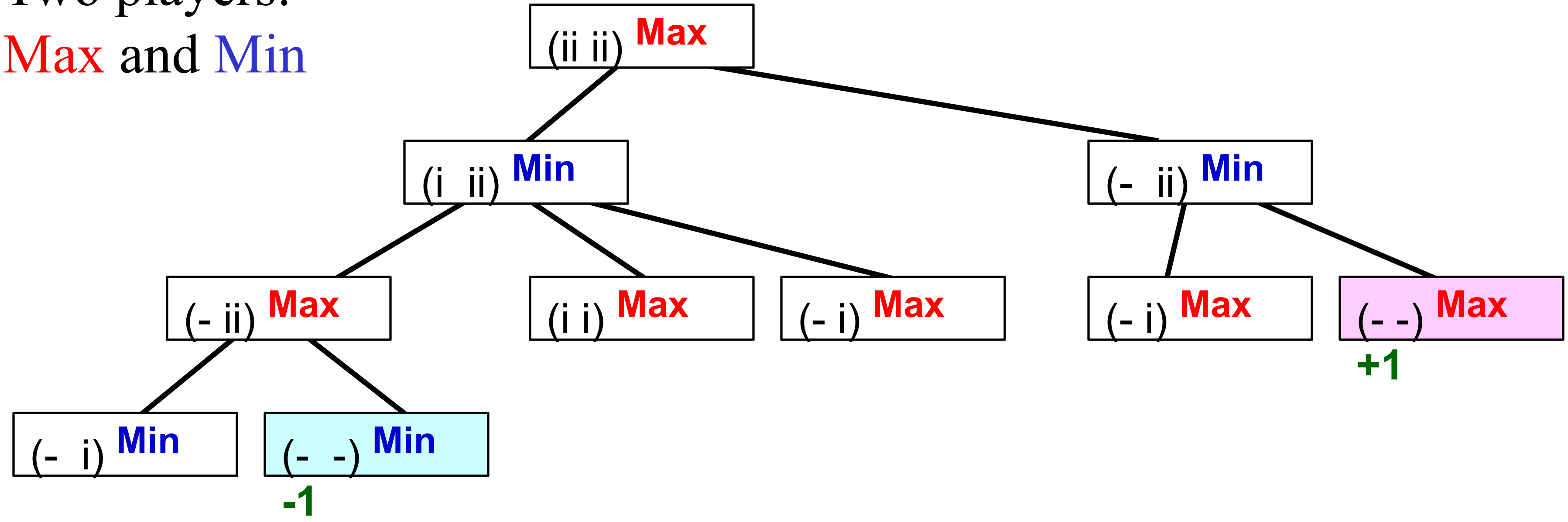
Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

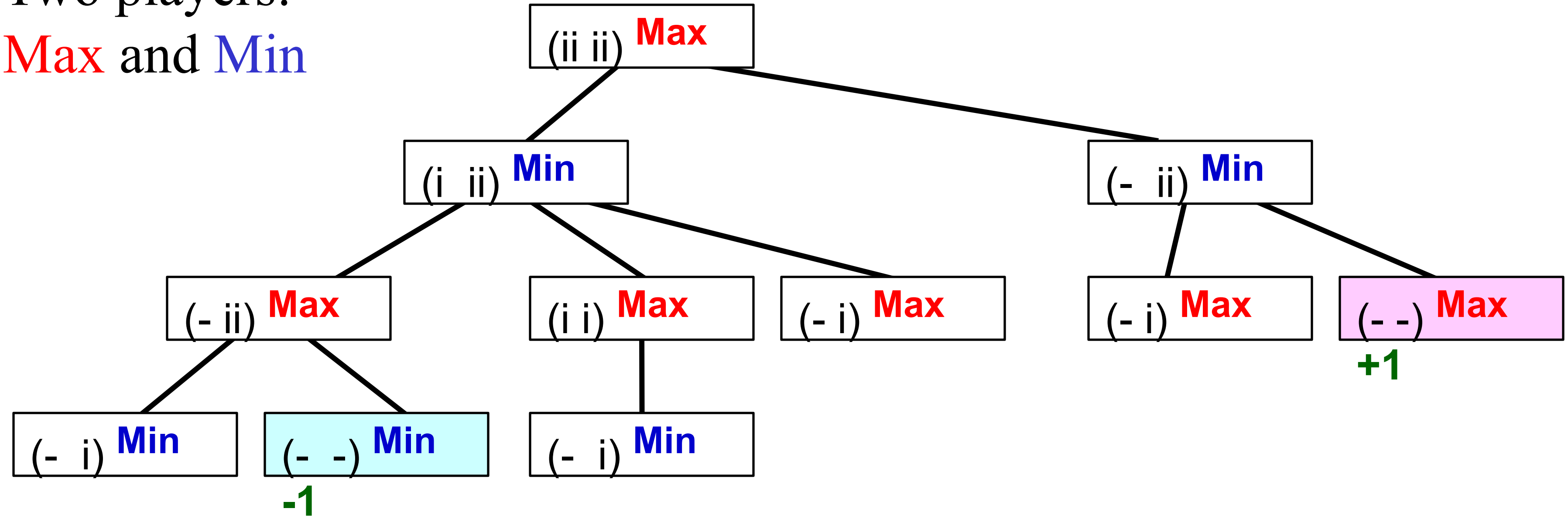
Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

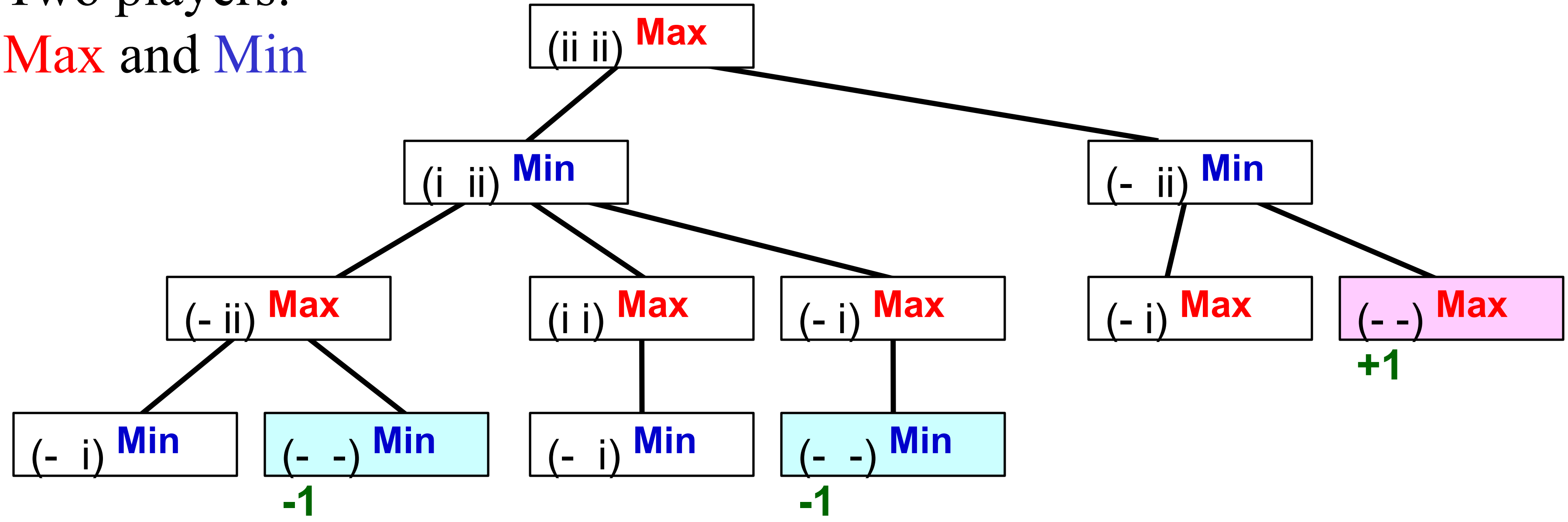
Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

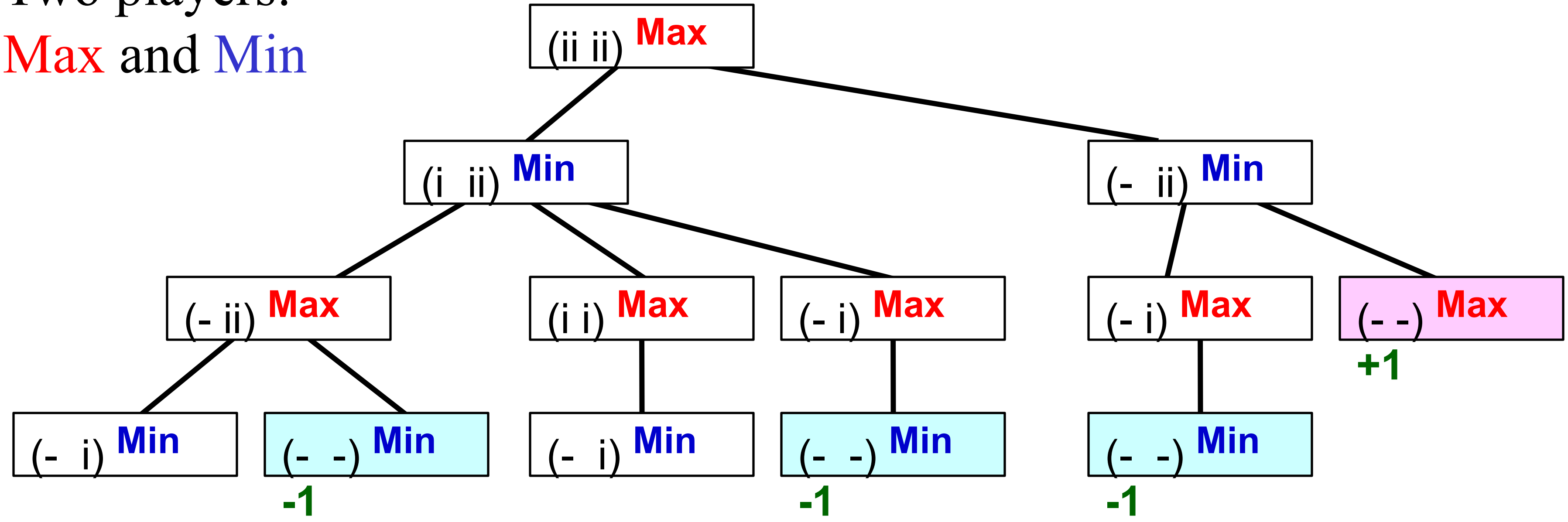
Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

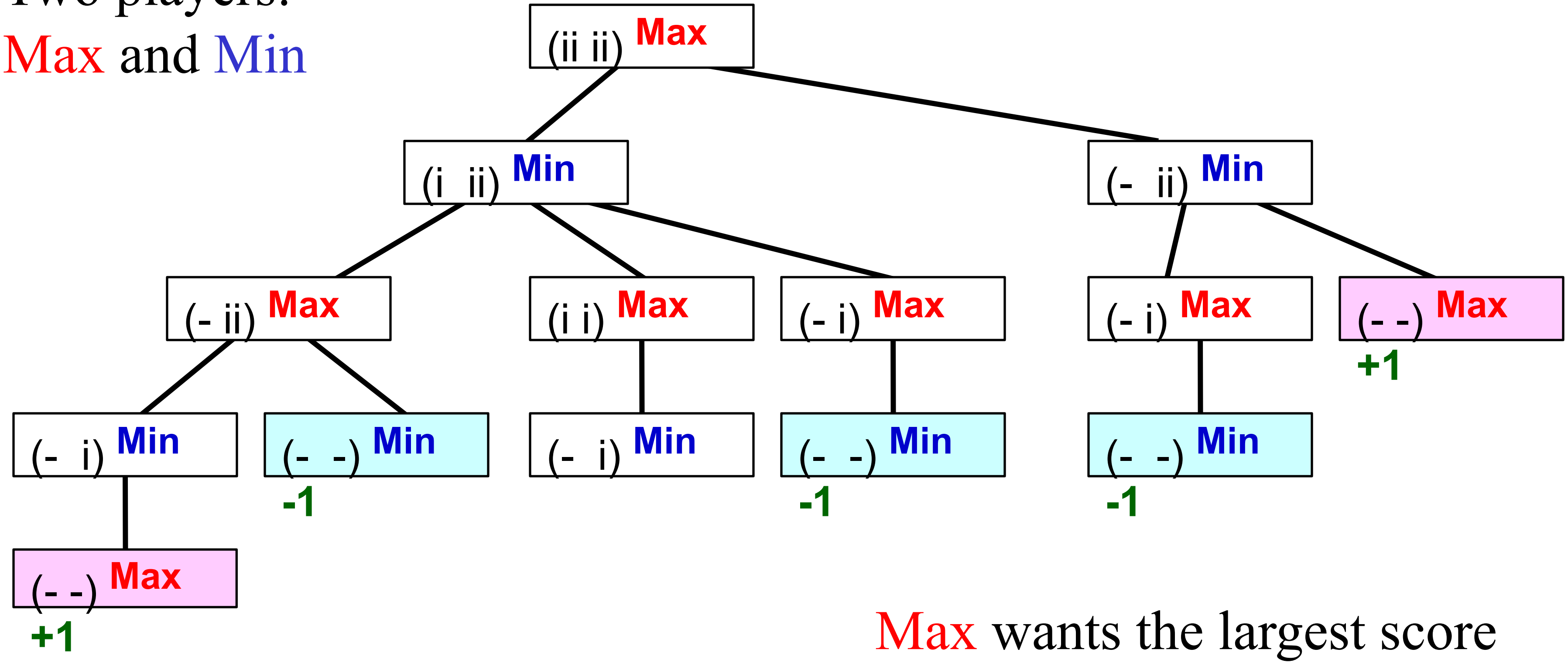
Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

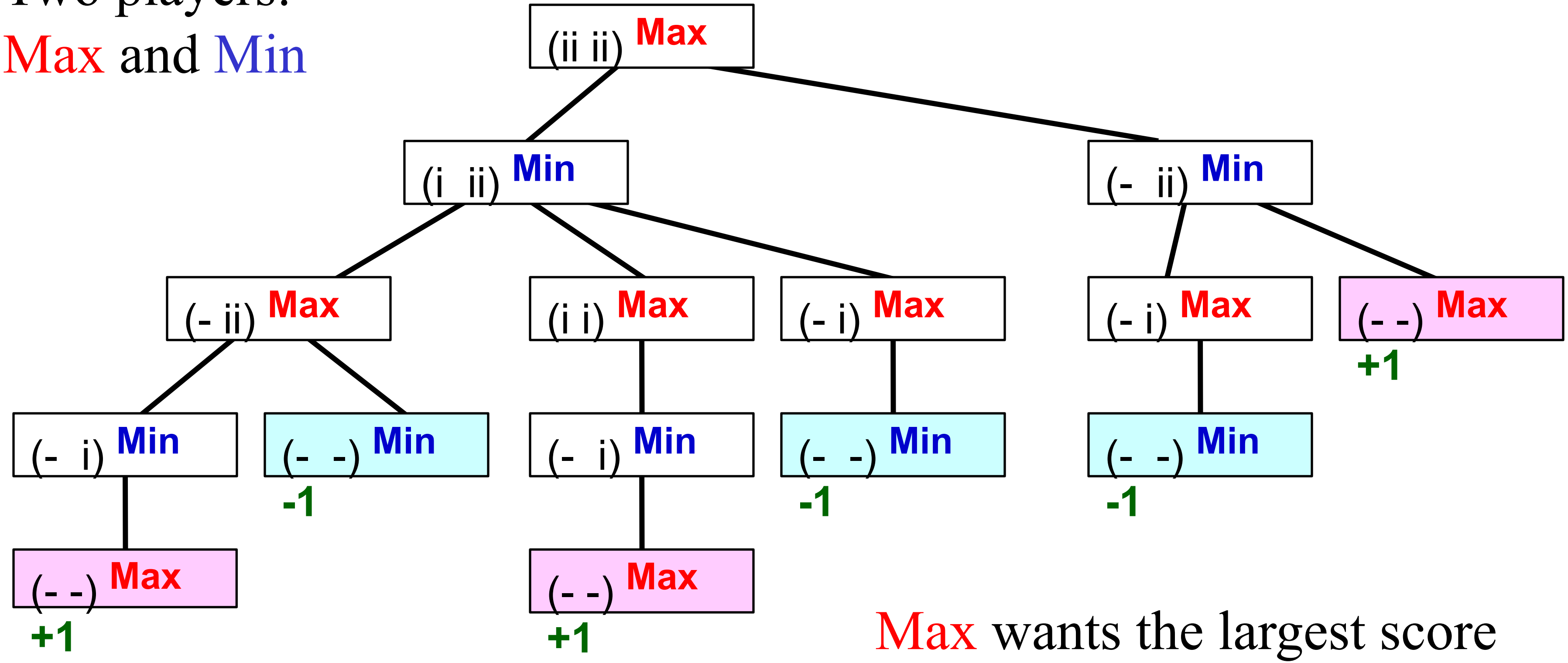
Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Minimax Value

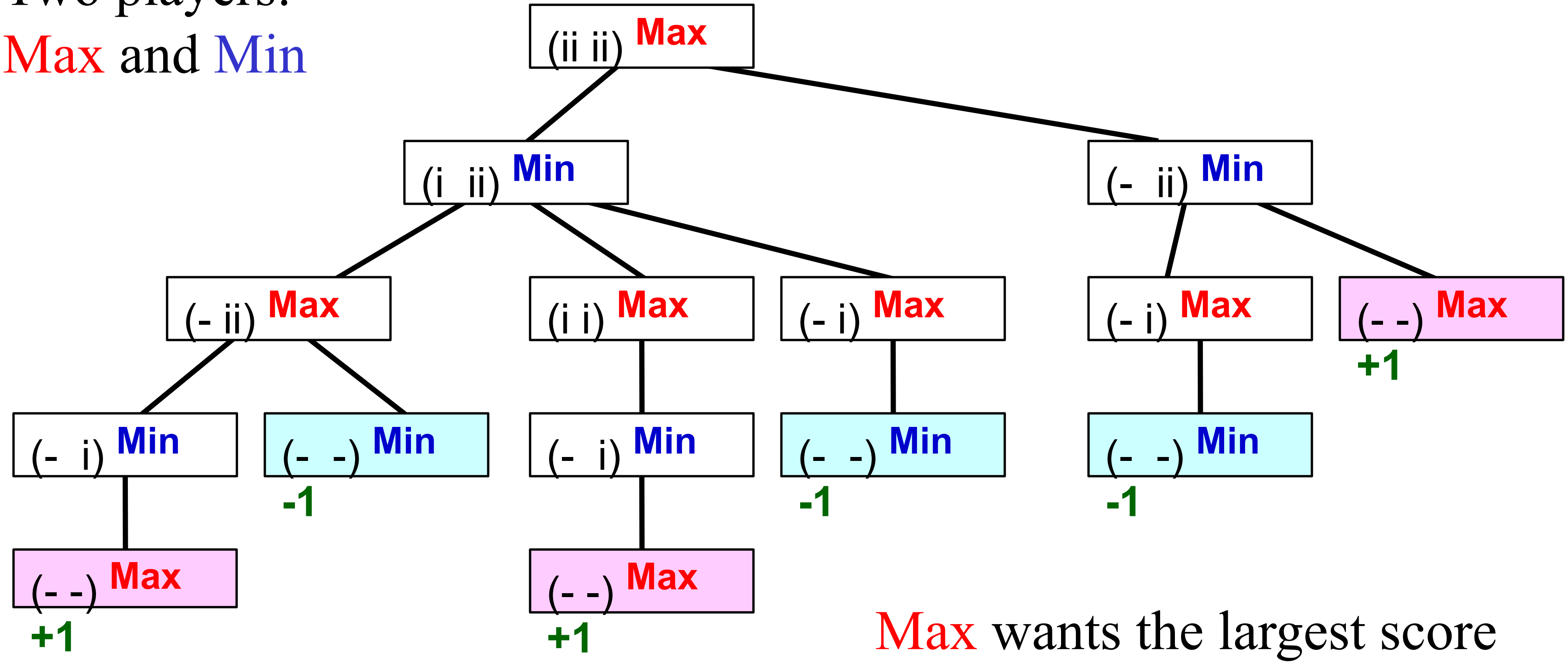
Also called **game-theoretic value**.

- Score of terminal node if both players play optimally.
- Computed bottom up; basically search

- Let’s see this for example game

Game tree for II-Nim

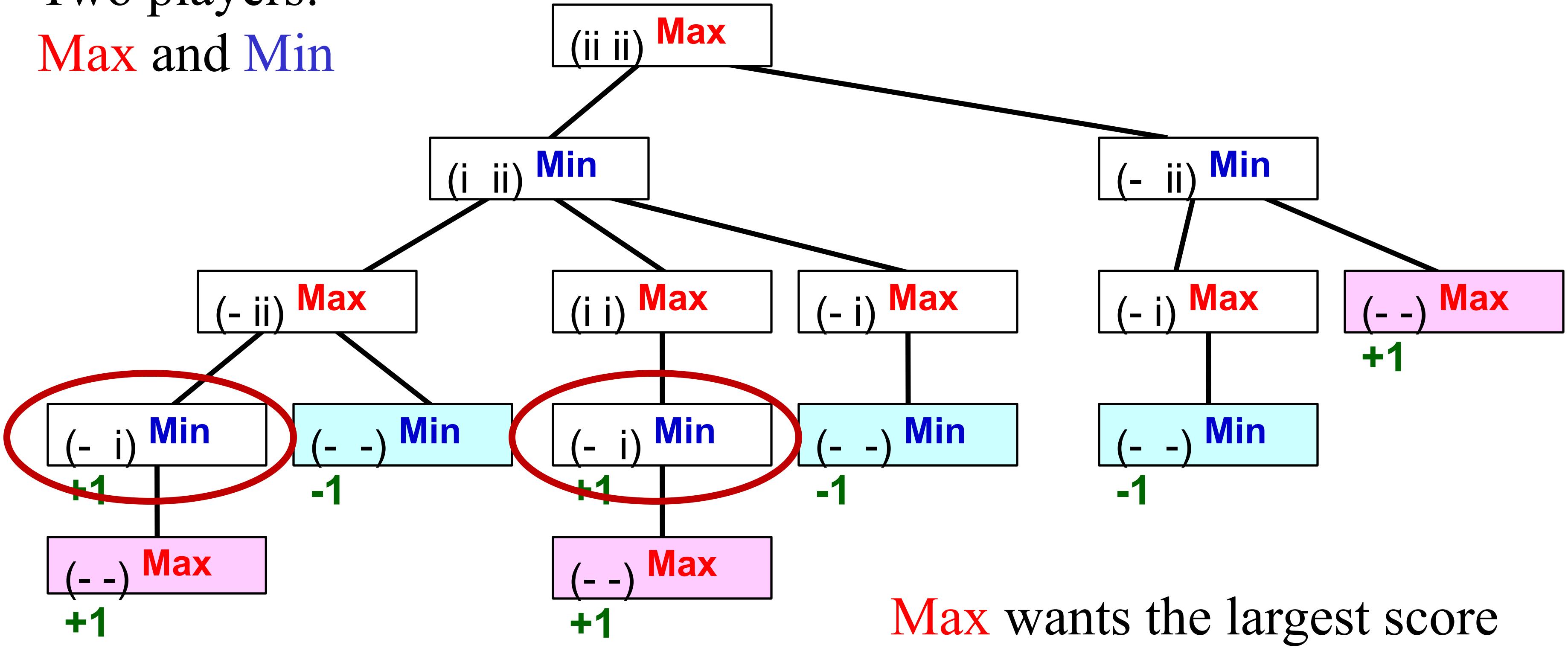
Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

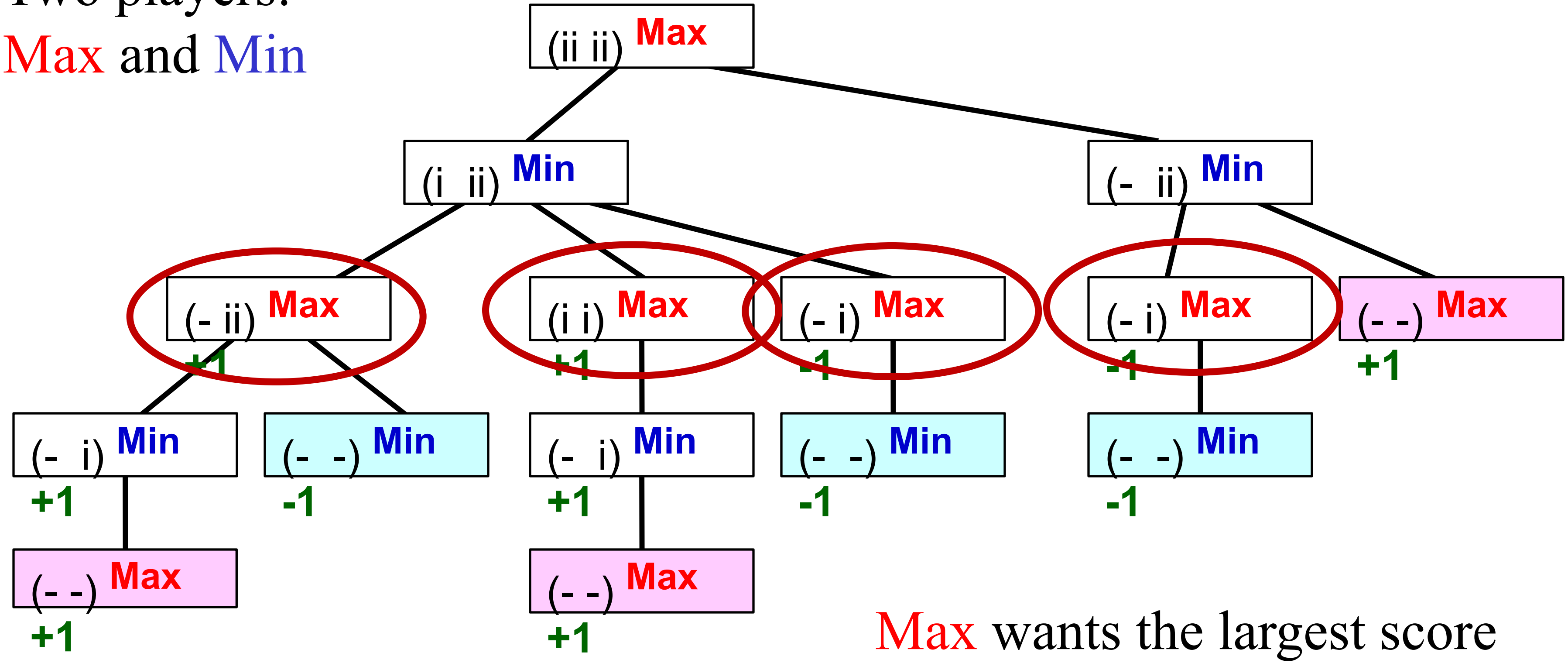
Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

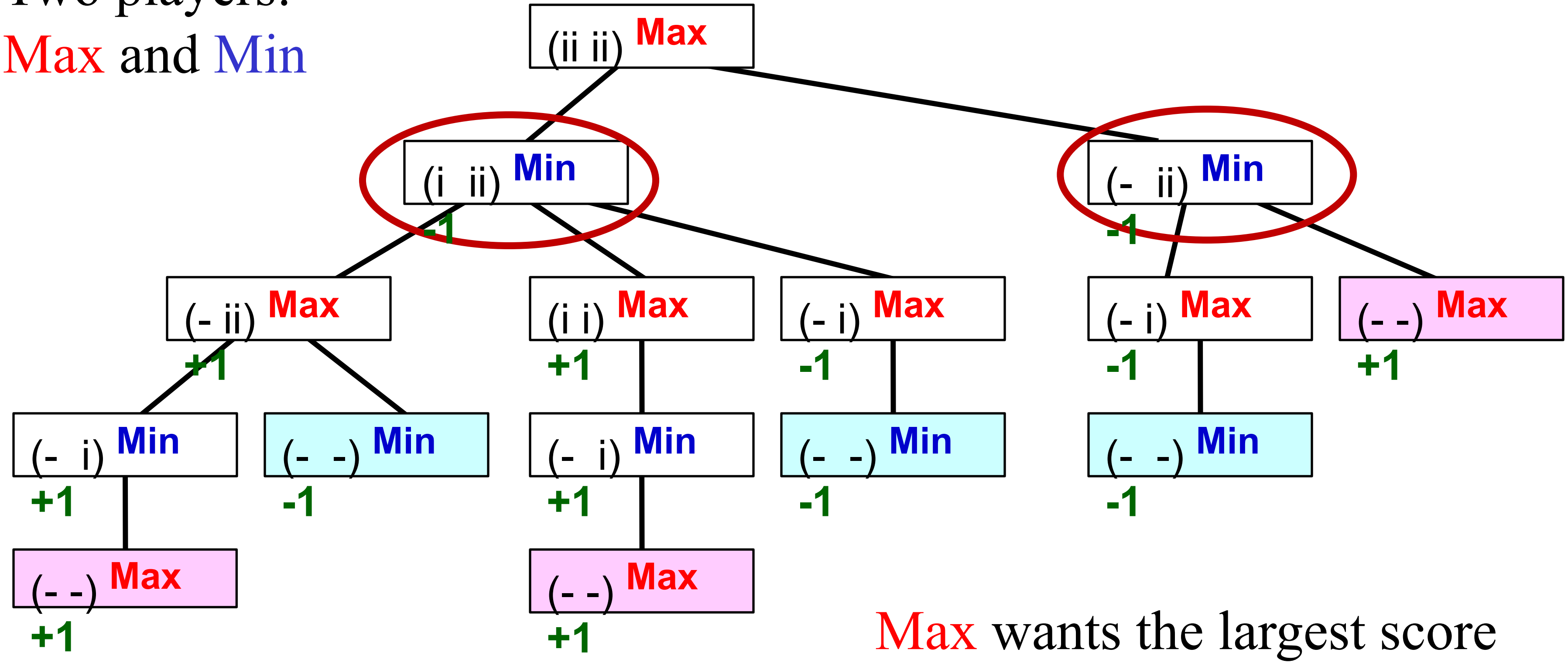
Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

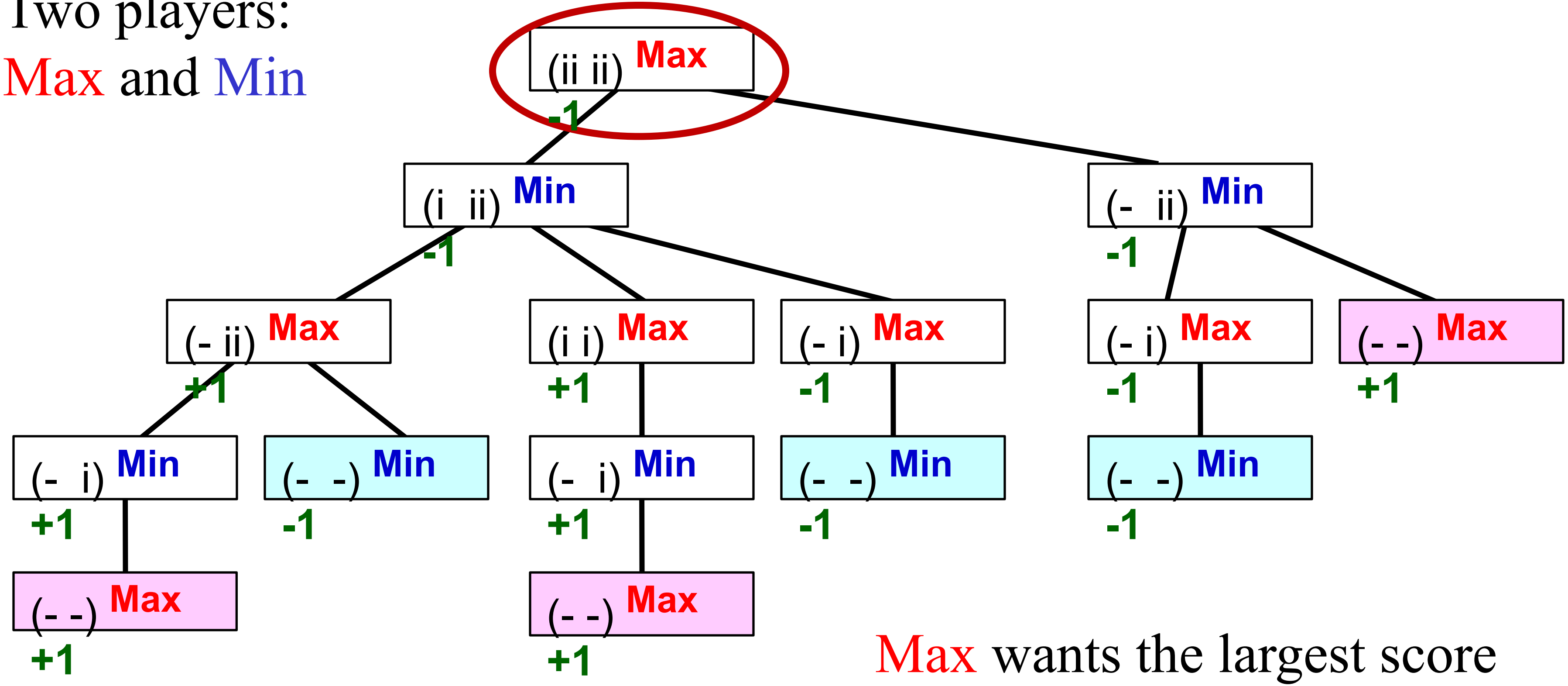
Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

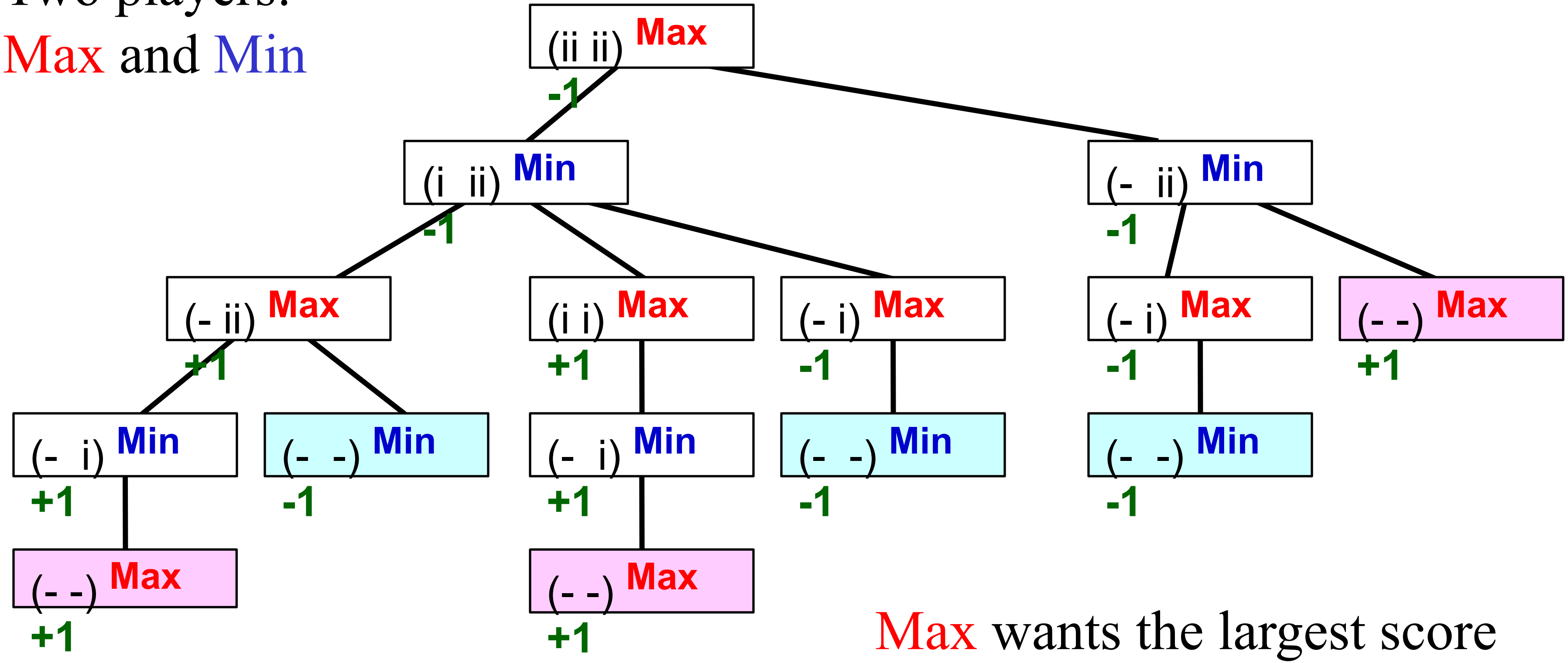
Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

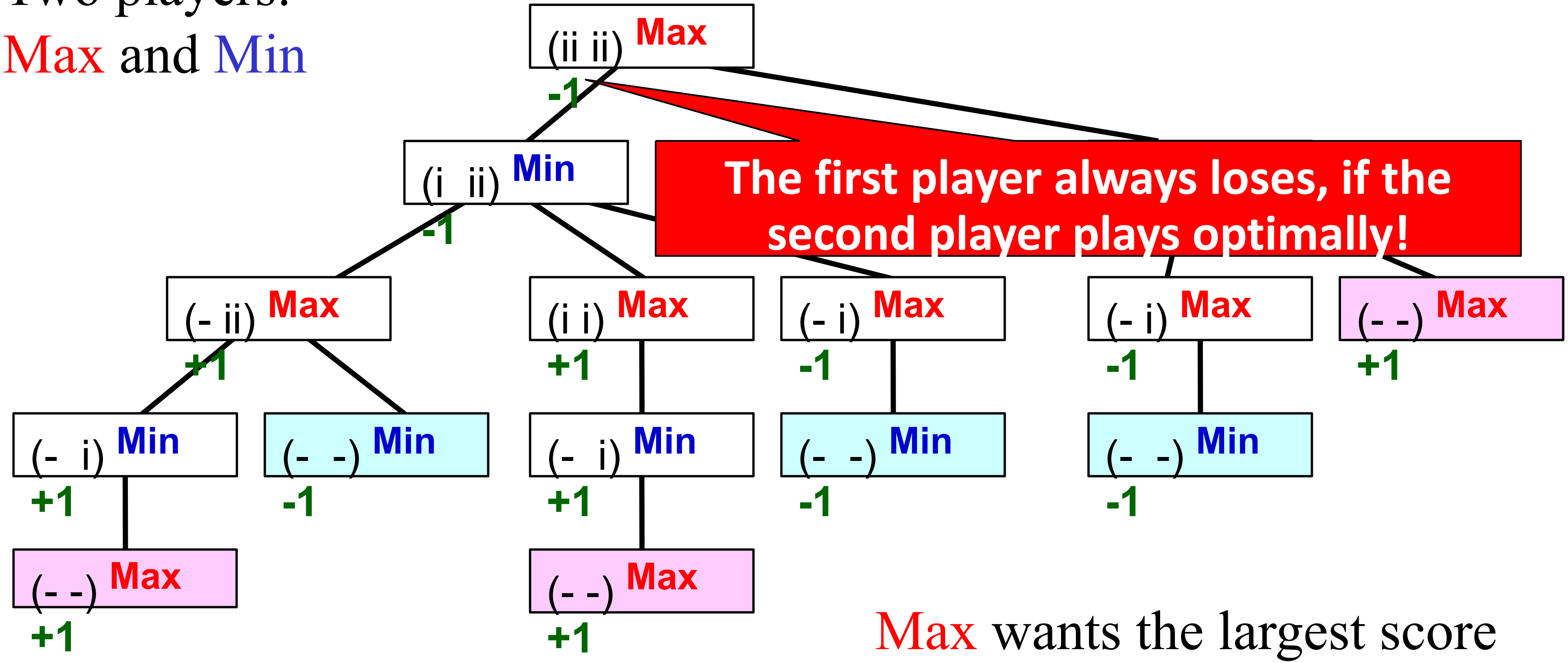
Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Game tree for II-Nim

Two players:
Max and Min



Max wants the largest score
Min wants the smallest score

Summary

- Intro to game theory
  - Characterize games by various properties
- Mathematical formulation for simultaneous games
  - Normal form, dominance, equilibria, mixed vs pure
- Sequential games
  - Game trees, game-theoretic/minimax value

Acknowledgements: Developed from materials by Yingyu Liang (University of Wisconsin), inspired by Haifeng Xu (UVA).